(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 682 327 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **95106967.3**

(51) Int. Cl.[6]: **G07F 7/10**, H04L 9/32

(22) Date of filing: **09.05.95**

(30) Priority: **09.05.94 US 240016**

(43) Date of publication of application:
15.11.95 Bulletin 95/46

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(71) Applicant: **YEDA RESEARCH AND DEVELOPMENT COMPANY, LTD.**
The Weizmann Institute of Science
76 100 Rehovot (IL)

(72) Inventor: **Shamir, Adi**
43C Gordon Street
Rehòvot (IL)

(74) Representative: **Laufhütte, Dieter, Dr.-Ing. et al**
Lorenz-Seidler-Gossel
Widenmayerstrasse 23
D-80538 München (DE)

(54) Method and apparatus for memory efficient variants of public key encryption and identification schemes for smart card applications.

(57) Memory efficient variants of public key encryption and identification schemes for smart card applications with severely limited RAM without using dedicated coprocessors. The variants replace the memory-intensive modular multiplication operation $z = x^*y \pmod n$ by a new randomized multiplication operation $z' = x^*y + r^*n$, where r is a randomly chosen integer in a suitable range [0,b], and a double convolution process to compute z'is used. Method and apparatus are described.

EP 0 682 327 A2

Field of the Invention

The invention relates to a method and an apparatus for public key encryption and identification schemes for smart card applications.

Background of the Invention

Almost all the public key encryption and identification schemes proposed so far are based on modular multiplications with a modulus n which is the product of two secret primes p and q. To make the factorization of n difficult, it is necessary to use very large numbers. The minimum recommended size of n is currently 512 bits, but due to the explosive growth of computing power available to cryptanalysts, this minimum size is likely to increase to 1024 bits (and to even larger values for high security applications).

In many communication and access control applications, it is desirable to use smart cards to carry out sensitive computations. Among the many reasons for this choice are the small physical size, the portability, the non-volatile memory, and the security offered by a single chip computer embedded in a plastic card. Millions of smart cards are used each year to make bank cards more secure, to control access to pay-TV, to carry billing information in cellular telephones, etc.

The biggest limitation in today's smart cards is the small amount of random access memory (RAM) available in the card. The most popular smart card chip made by Motorola has 36 bytes of RAM, and the most popular smart card chip made by Thomson has 44 bytes of RAM. This should be compared with the 4 million bytes of RAM available in a typical personal computer.

Some algorithms can make use of other types of memory: A typical smart card contains several thousand bytes of ROM and several thousand bytes of EEPROM. ROM is unchangeable, and typically stores the program which controls the operation of the smart card. EEPROM is changeable, but writing into it is about one thousand times slower than writing into RAM and the number of times a bit can be rewritten cannot exceed 10,000. It is thus possible to use EEPROM to store slowly changing data such as cryptographic keys or the details of financial transactions, but not as a RAM substitute for intermediate values in a long computation.

Summary of the Invention

The present invention relates to a method and apparatus for modifying two of the most popular public key schemes (the Rabin encryption scheme and the Fiat-Shamir identification scheme) in order to make them suitable for smart cards with severely limited RAMs. The modified variants are as secure as the original variants, but they are so space-efficient that even the simplest 36 byte smart card can handle moduli n with thousands of bits without any difficulty.

It is an object of the invention to provide a method and apparatus for a public key scheme for smart cards with severely limited RAM without using dedicated coprocessors.

According to the present invention the novel memory efficient variants of public key encryption and identification schemes for smart card applications can be accomplished using as few as 36 bytes of RAM. A new randomized multiplication operation $z' = x*y + r*n$ is used where r is a randomly chosen integer in a suitable range [0,b]. A double convolution process is used to compute z'.

Brief Description of the Figures

Fig. 1 shows schematically a smart card as used in the invention.

Fig. 2 shows schematically a memory efficient variant of a Rabin scheme as implemented in a smart card according to the invention.

Fig. 3 shows schematically a memory efficient variant of a Fiat-Shamir scheme as implemented in a smart card according to the invention.

Detailed Description of the Preferred Embodiments

First one considers the basic operation of multiplying two large numbers x and y which are already stored in a smart card 10 (e.g., in its EEPROM 12), see Fig. 1. The successive bytes of x and y are denoted by x_i and y_i (respectively) for $i = o,...,k-1$, where the i-th byte is the least significant one. Smart card 10 has a ROM 14, a RAM 16, a microprocessor chip 18 and the usual I/O 20. The result $z = x*y$ cannot fit in the small RAM, but if it is the final result computed by the smart card, its successive bytes can be sent

out (rather than stored) as soon as they are generated.

According to the present invention convolution is used for multiplying two k-byte numbers in $O(k^2)$ time with O(log(k)) workspace. Start with c = 0. To compute the i-th byte of the result z for i = 0, 1, 2,...k-1, compute:

$$t = c + \sum_{j=0}^{i} (x_j * y_{i-j})$$

for j = 0, 1, 2,...i, send the least significant byte of t as z_i, and use the value of the other bytes of t as the new carry c. Note that for any x and y with up to half a million bits, t fits into 4 bytes, and thus the algorithm can be easily implemented even on the cheapest smart card with 36 bytes of RAM.

Next the problem of computing z = x*y (mod n) is considered. By definition, z = x*y - w*n where

$$w = \lfloor x*y/n \rfloor$$

(i.e., w is x*y/n truncated to the largest integer below it). Since z cannot be stored, it is not obvious how to carry out this division operation. One can try to generate the successive bytes of z by the convolution method, but in the division operation one would need these bytes from left to right, whereas the convolution computes them from right to left. Thus one would be forced to recalculate each byte of z a large number of times, and the cryptographic scheme becomes unacceptably slow.

The new solution to this problem according to the invention is to replace the modular multiplication operation z = x*y - w*n where

$$w = \lfloor x*y/n \rfloor$$

by a new randomized multiplication operation z' = x*y + r*n where r is a randomly chosen integer in a suitable range [0,b]. Such a z' can be easily computed by the following double convolution process:

1. Set c = 0.
2. For i = 0, 1,...k-1 compute:

$$t = c + \sum_{j=0}^{i} (x_j * y_{i-j}) + \sum_{m=0}^{i} (r_m * n_{i-m})$$

3. Send the low-order byte of t as z_i, and
4. Set c to the number represented by the other bytes of t.

Clearly, the value of z can be recovered from the value of z' by reducing z' modulo n, and thus there is no loss of information in sending z' instead of z. Now it will be shown that when b is a large enough public bound, there is no loss of cryptographic security in sending z' instead of z. Assume that there exists an attack against the cryptographic scheme which uses the knowledge of z' = x*y + r*n for a random r in [0,b]. These values define a probability space P'. We show that such an attack can also be based on the deterministic value of z = x*y - w*n. Given z, the cryptanalyst can easily compute by himself another probability space P'' defined by the values z'' = z + u*n for a random u in [0,b]. By definition, z'' = x*y + (u-w)*n. The only difference between P' and P'' is that the range of coefficients of n is shifted by the unknown quantity w from [0,b] to [-w, b-w]. The probability that a random point in one range will fall outside the other is w/b. If x and y are in [0, n], and b is much larger than n, then this probability is negligible (e.g., when b is a power of 2 which is 64 bits longer than n, this probability is less than 2^{-64}). Since P' and P'' are statistically indistinguishable, any cryptanalytic attack will be equally successful over the two probability spaces.

The only disadvantage of randomized multiplication with respect to modular multiplication is that the transmitted result is about twice as long. However, this adds only a negligible communication delay, and the recipient computer (which is usually a powerful PC or a network server) can immediately change z' to z before storing or processing it further.

In the next two sections is described how to use randomized multiplications in order to obtain space-efficient variants of the Rabin and Fiat-Shamir cryptographic schemes.

A Space-Efficient Variant of the Rabin Encryption Scheme

Rabin's public key scheme is used to establish a common secret key K between two parties, which can then be used to encrypt the actual message with a standard cryptosystem such as DES. K is usually the low order bits (56 in the case of DES) of a long number x in the range [0,n] chosen randomly by the sender. However, K can be defined as any other publicly known function of x. The sender then computes and sends $z = x^*x \pmod n$, and the recipient uses his knowledge of the factorization of n in order to compute the modular square root of z (mod n). A slight complication is that z yields four possible K's, but this can be solved by known techniques.

Referring now to Fig. 2 in the proposed new variant, K is sent by computing the randomized multiplication $z' = x^*x + r^*n$ instead of the modular multiplication $z = x^*x \pmod n$. In the preferred implementation of the new variant, n (which is the product of two large primes p and q) is kept in EEPROM, see block 30. It is usually the public key of the organization which issues the cards to its employees and customers, and even small cards with one kilobyte of EEPROM can store an 8000 bit modulus. The numbers x and r are pseudo randomly generated from random secret seeds sx and sr (which are loaded into EEPROM when the card is issued) see block 32, a session counter v (which is incremented in EEPROM at the beginning of each communication session) see block 34, and a byte index j, see block 36. A convenient way of achieving this is to hash sx, v, and j into the j-th byte of x in session v, see block 38, and to hash sr, v, and j into the j-th byte of r in session v, see block 40. In this way it is possible to access individual bytes of x and r in any order without storing x and r anywhere, and thus it is possible for a prover to compute, see block 42, and send out the successive bytes of z', see block 44, by the double convolution method even when n has thousands of bits and the card has only 36 bytes of RAM. The successive bytes are received by a verifier, see block 46 and verification is effected, see block 48.

A Space-Efficient Variant of the Fiat-Shamir Identification Scheme

The Fiat-Shamir identification scheme is described in U.S. Patents Nos. 4,748,668 and 4,993,970, the disclosures of which are incorporated herein by reference. In this section we describe a novel space-efficient variant of this scheme which makes it possible to implement it on smart cards with very small RAMs.

In the original Fiat-Shamir identification scheme, the smart card (known as the prover) contains in its EEPROM a public modulus $n = p^*q$ and a secret number c. The other party (known as the verifier) knows n and $d = c^*c \pmod n$. The smart card proves its identity to the verifier by using a zero knowledge protocol to demonstrate its knowledge of c. The proof consists of the following steps:

1. The prover chooses a random number x, and sends $z = x^*x \pmod n$ to the verifier.
2. The verifier sends a random bit to the prover.
3. Based on the bit, the prover sends either x or $x^*c \pmod n$ to the verifier.
4. Based on the bit, the verifier checks that the square of the received number is either z (mod n) or $z^*d \pmod n$.
5. Steps 1-4 are repeated several times to reduce the probability of cheating.

In the new space-efficient variant of the Fiat-Shamir scheme, the prover performs the same steps, but replaces the modular multiplications $x^*x \pmod n$ and $x^*c \pmod n$ by the randomized multiplications $x^*x + r^*n$ and $x^*c + t^*n$ for pseudo-random r and t in [0,b], where b is substantially larger than n (e.g., by at least 64 bits). As demonstrated above, these operations can be carried out with very small RAMs.

Referring to Fig. 3, in block 80, n is stored as the product of large primes p and q, in block 82 pseudo-random v and t are stored, in block 84 the first computation is made and sent to the verifier who receives in block 86 and then sends random bit a number of times for verifying calculations in blocks 88-98. Finally, a verification is made in block 100. The process may be carried out using the double convolution techniques described in the foregoing.

While the invention has been described with respect to certain embodiments thereof, it will be appreciated by one skilled in the art that variations and modifications may be made without departing from the spirit and scope of the invention.

Claims

1. A method of computing in a space-efficient and cryptographically-secure way a randomized variant of the modular product $z = x^*y \pmod n$ comprising the steps of:
   (a) storing two k-byte numbers x and y in memory of a smart card including memory, a microprocessor and I/O,
   (b) computing the i-th byte of the result of $z' = x^*y + r^*n$ where y is a randomly chosen integer in a range [o,b] and n is the product of two large primes p and q, using double convolution by
      (i) setting $c = 0$
      (ii) for $i = 0, 1...k-1$, computing

$$t = c + \sum_{j=0}^{i} (x_j * y_{i-j}) + \sum_{m=0}^{i} (r_m * n_{i-m})$$

   (c) transmitting the low-order byte of t as z_i via the I/O of the smart card,
   (d) repeating the computation and transmission of steps (b) and (c) using the number represented by the other bytes of t for c.

2. A method of claim 1 including the further steps of:
   (e) receiving the transmitted bytes, and
   (f) recovering the product of the two k-byte numbers x and y (modulo n) from the received bytes.

3. A method of encryption of a secret key K comprising the steps of:
   (a) storing a public key n (the product of two large primes p and q) in a EEPROM of a smart card having EEPROM, RAM, a microprocessor and I/O,
   (b) storing secret seeds sx and sr in the EEPROM,
   (c) establishing a session counter v and a byte index j in the EEPROM,
   (d) incrementing the session counter at the beginning of each communications session,
   (e) hashing sr, v and j and also sx, v and j into the j-th byte of r and x, respectively, in session v,
   (f) computing successive bytes of $z' = x^*x + r^*n$ using the method of claim 1,
   (g) transmitting via the I/O successive bytes of z',
   (h) defining the secret key K as some publicly known function of x.

4. A method of identification in which an entity claims that it knows a secret value c related to the public values d and n by $d = c^*c \pmod n$, and proves it by using the steps of:
   (a) storing c and n (which is the product of two large primes p and q) in an EEPROM of a smart card having EEPROM, RAM, a microprocessor and I/O,
   (b) storing a secret seed sx and sn in the EEPROM,
   (c) establishing a multiplication counter v and a byte index j in the EEPROM,
   (d) incrementing the multiplication counter at the beginning of each modular multiplication in each proof of identity,
   (e) hashing sn, v and j and also sx, v and j into the j-th byte of r and x, respectively, in session v,
   (f) computing successive bytes of $z' = x^*x + r^*n$ using the method of claim 1,
   (g) transmitting via the I/O successive bytes of z',
   (h) receiving from the verifier a random bit,
   (i) based on the bit, sending either x or a randomized version of $x^*c \pmod n$, according to the method of claim 1,
   (j) repeating steps (d) to (i) one or more times to reduce the probability of cheating.

5. Apparatus for computing in a space-efficient and cryptographically-secure way a randomized variant of the modular product $z = x^*y \pmod n$ comprising:
   (a) means for storing two k-byte numbers x and y in memory of a smart card including memory, a microprocessor and I/O,
   (b) means for computing the i-th byte of the result of $z' = x^*y + r^*n$ where y is a randomly chosen integer in a range [o,b] and n is the product of two large primes p and q, using double convolution by

(i) setting c = 0
(ii) for i = 0, 1...k-1, computing

$$t = c + \sum_{j=0}^{i} (x_j * y_{i-j}) + \sum_{m=0}^{i} (r_m * n_{i-m})$$

(c) means for transmitting the low-order byte of t as z_i via the I/O of the smart card,
(d) means for repeating the computation and transmission of steps (b) and (c) using the number represented by the other bytes of t for c.

6. Apparatus of claim 5 further comprising:
(e) means for receiving the transmitted bytes, and
(f) means for recovering the product of the two k-byte numbers x and y (modulo n) from the received bytes.

7. Apparatus for encryption of a secret key K comprising:
(a) means for storing a public key n (the product of two large primes p and q) in a EEPROM of a smart card having EEPROM, RAM, a microprocessor and I/O,
(b) means for storing secret seeds sx and sr in the EEPROM,
(c) means for establishing a session counter v and a byte index j in the EEPROM,
(d) means for incrementing the session counter at the beginning of each communications session,
(e) means for hashing sr, v and j and also sx, v and j into the j-th byte of r and x, respectively, in session v,
(f) means for computing successive bytes of z' = x*x + r*n using the apparatus of claim 5,
(g) means for transmitting via the I/O successive bytes of z',
(h) means for defining the secret key K as some publicly known function of x.

8. Apparatus for identification in which an entity asserts that it knows a secret value c related to the public values d and n by d = c*c (mod n), and proves it by using apparatus comprising:
(a) means for storing c and n (which is the product of two large primes p and q) in an EEPROM of a smart card having EEPROM, RAM, a microprocessor and I/O,
(b) means for storing a secret seed sx and sn in the EEPROM,
(c) means for establishing a multiplication counter v and a byte index j in the EEPROM,
(d) means for incrementing the multiplication counter at the beginning of each modular multiplication in each proof of identity,
(e) means for hashing sn, v and j and also sx, v and j into the j-th byte of r and x, respectively, in session v,
(f) means for computing successive bytes of z' = x*x + r*n using the apparatus of claim 5,
(g) means for transmitting via the I/O successive bytes of z',
(h) means for receiving from the verifier a random bit,
(i) means for sending either x or a randomized version of x*c (mod n), based on the bit, using the apparatus of claim 5, and
(j) means for repeating steps (d) to (i) one or more times to reduce the probability of cheating.

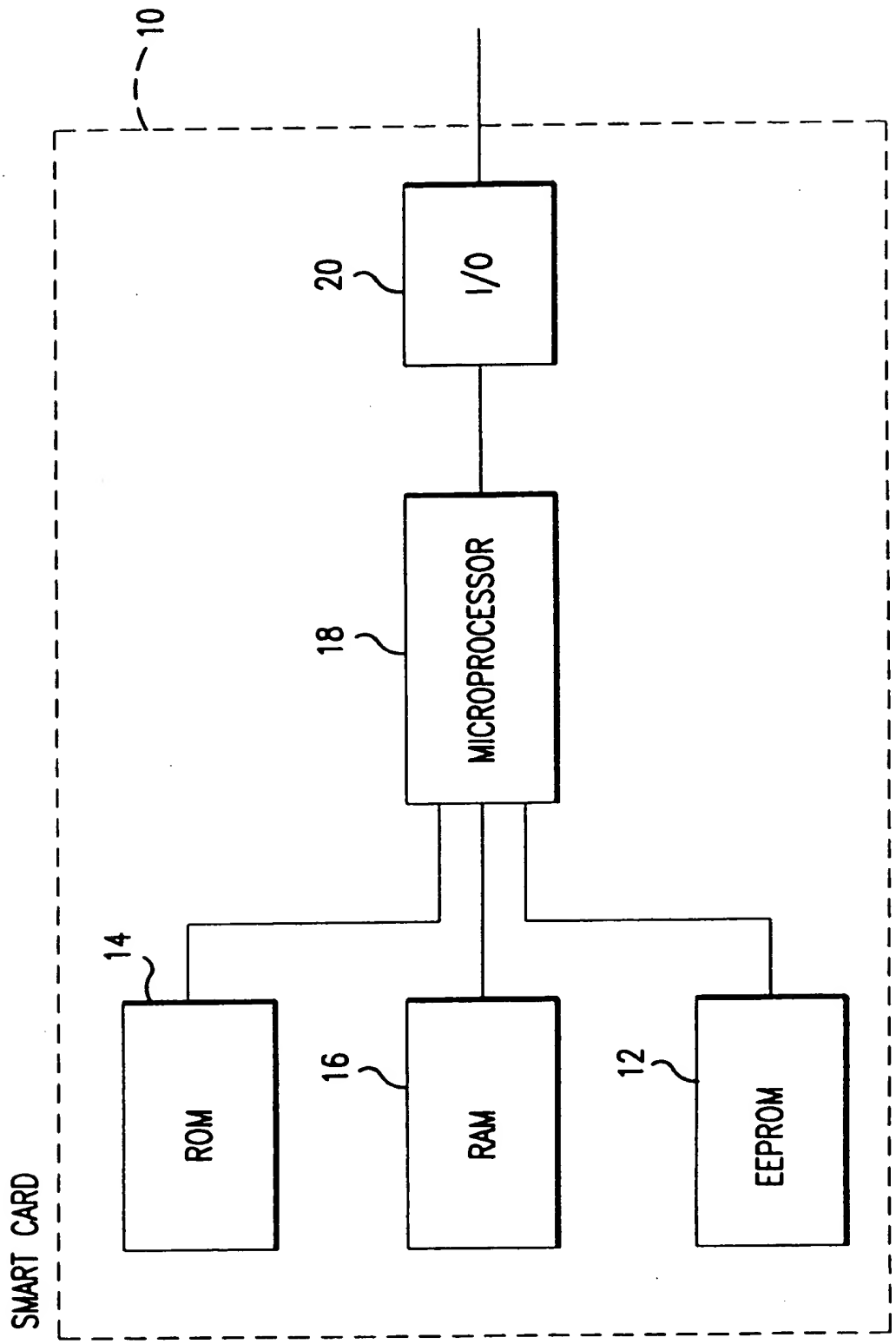

FIG.1

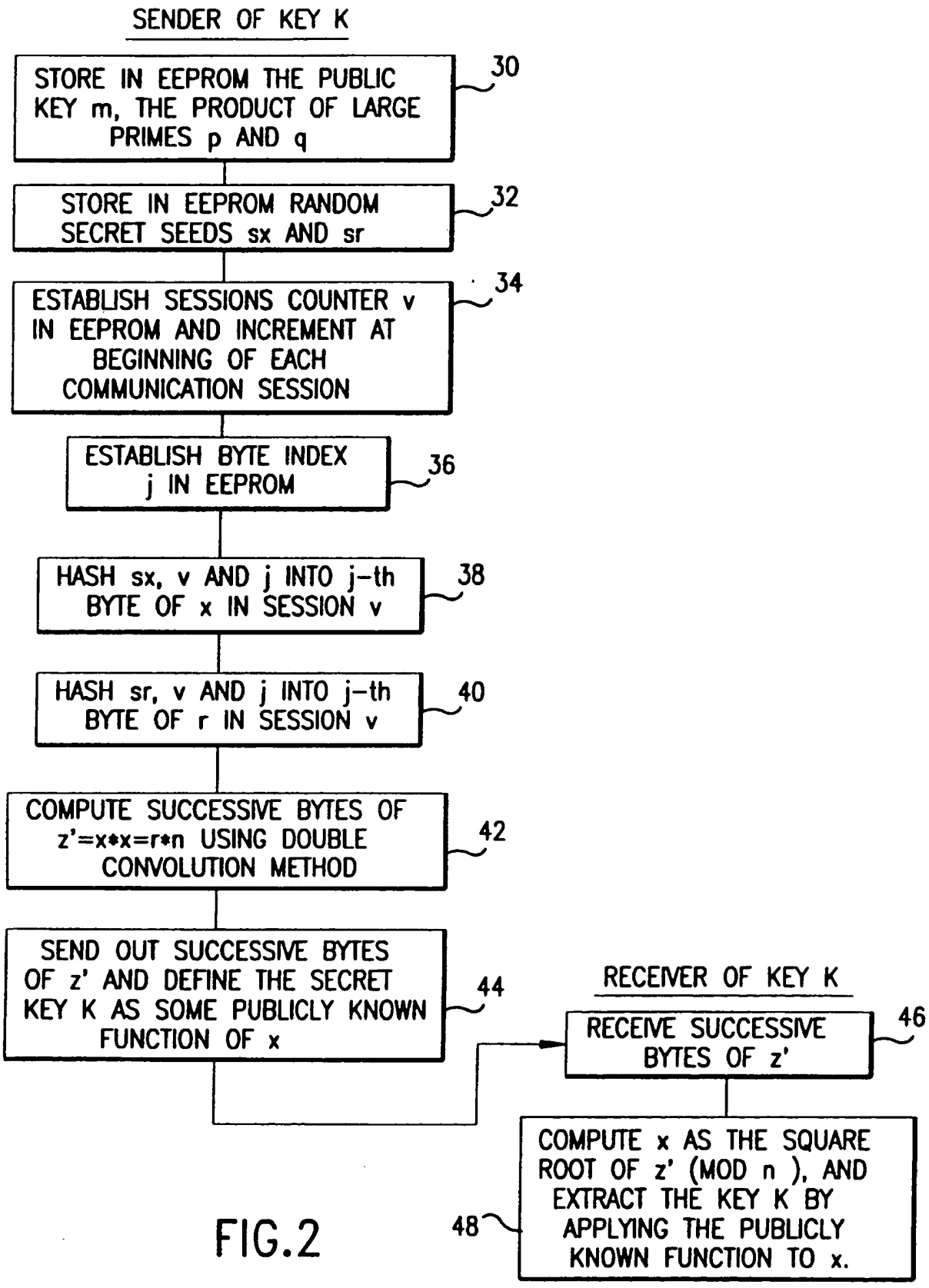
SENDER OF KEY K
STORE IN EEPROM THE PUBLIC KEY m, THE PRODUCT OF LARGE PRIMES p AND q
30
STORE IN EEPROM RANDOM SECRET SEEDS sx AND sr
32
ESTABLISH SESSIONS COUNTER v IN EEPROM AND INCREMENT AT BEGINNING OF EACH COMMUNICATION SESSION
34
ESTABLISH BYTE INDEX j IN EEPROM
36
HASH sx, v AND j INTO j-th BYTE OF x IN SESSION v
38
HASH sr, v AND j INTO j-th BYTE OF r IN SESSION v
40
COMPUTE SUCCESSIVE BYTES OF z'=x*x=r*n USING DOUBLE CONVOLUTION METHOD
42
SEND OUT SUCCESSIVE BYTES OF z' AND DEFINE THE SECRET KEY K AS SOME PUBLICLY KNOWN FUNCTION OF x
44
RECEIVER OF KEY K
RECEIVE SUCCESSIVE BYTES OF z'
46
COMPUTE x AS THE SQUARE ROOT OF z' (MOD n), AND EXTRACT THE KEY K BY APPLYING THE PUBLICLY KNOWN FUNCTION TO x.
48

FIG.2

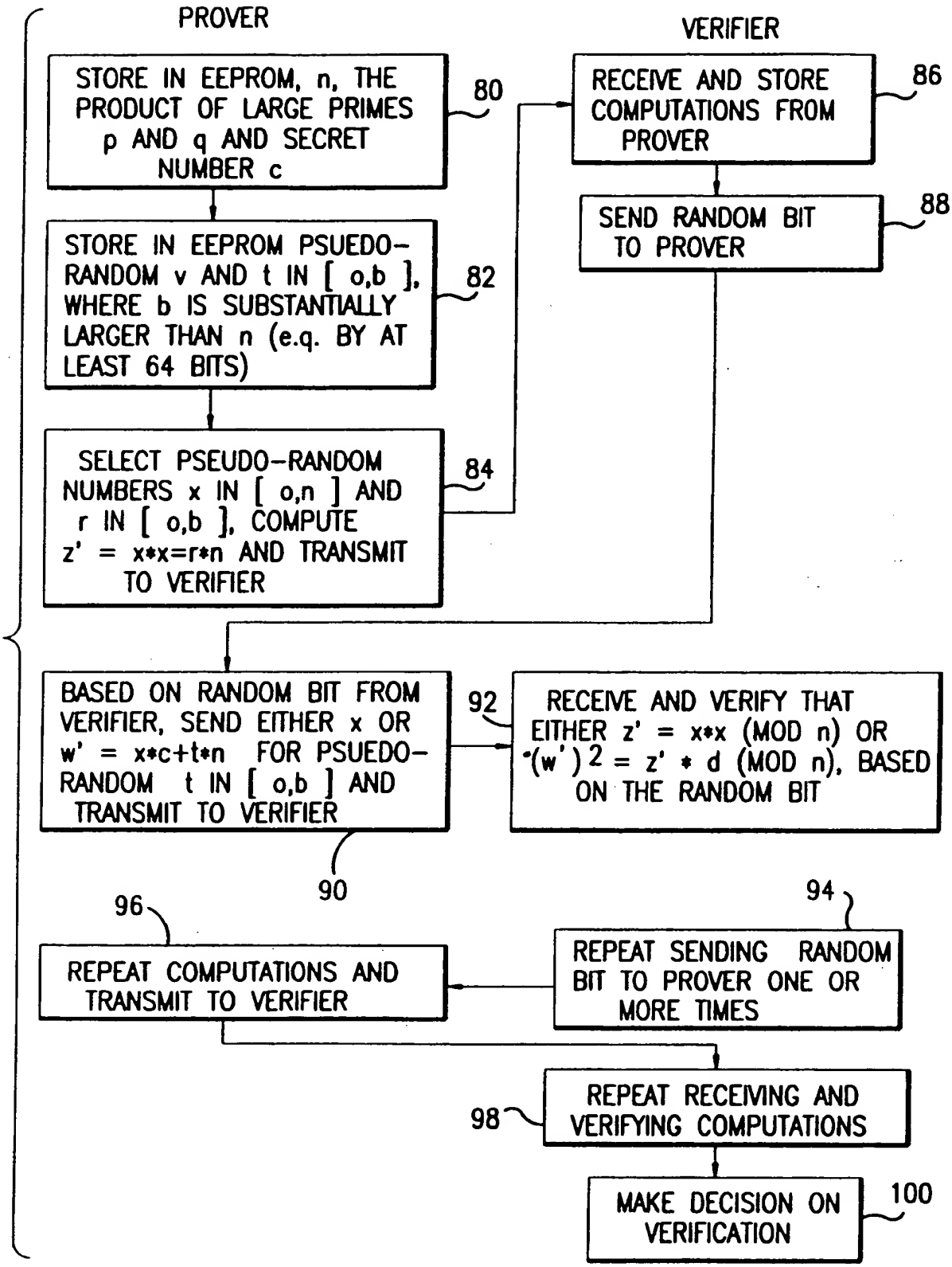

FIG.3

(19) European Patent Office
Office européen des brevets

(11) EP 0 682 327 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
03.11.1999 Bulletin 1999/44

(43) Date of publication A2:
15.11.1995 Bulletin 1995/46

(21) Application number: **95106967.3**

(22) Date of filing: **09.05.1995**

(51) Int. Cl.[6]: **G07F 7/10**, H04L 9/32

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **09.05.1994 US 240016**

(71) Applicant:
YEDA RESEARCH AND DEVELOPMENT COMPANY, LTD.
76 100 Rehovot (IL)

(72) Inventor: **Shamir, Adi**
Rehovot (IL)

(74) Representative:
Laufhütte, Dieter, Dr.-Ing. et al
Lorenz-Seidler-Gossel
Widenmayerstrasse 23
80538 München (DE)

(54) Method and apparatus for memory efficient variants of public key encryption and identification schemes for smart card applications

(57) Memory efficient variants of public key encryption and identification schemes for smart card applications with severely limited RAM without using dedicated coprocessors. The variants replace the memory-intensive modular multiplication operation $z = x^*y \pmod n$ by a new randomized multiplication operation $z' = x^*y + r^*n$, where r is a randomly chosen integer in a suitable range [0,b], and a double convolution process to compute z'is used. Method and apparatus are described.

EP 0 682 327 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 10 6967

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	WO 93 09620 A (THOMSON CONSUMER ELECTRONICS) 13 May 1993 * the whole document *	1-8	G07F7/10 H04L9/32
A,D	US 4 748 668 A (SHAMIR ADI ET AL) 31 May 1988 * column 2, line 54 - column 4, line 15 *	3,4,7,8	
A	US 5 121 431 A (WIENER MICHAEL J) 9 June 1992 * column 2, line 15 - column 4, line 11 * * column 5, line 6 - column 10, line 57 * * column 11, line 9 - column 11, line 40 *	1,2,5,6	
A	US 5 299 263 A (BELLER MICHAEL J ET AL) 29 March 1994 * column 1, line 17 - column 1, line 30 * * column 4, line 60 - column 6, line 6 * * column 10, line 6 - column 11, line 38 *	3,4,7,8	
A	EP 0 502 712 A (CANON KK) 9 September 1992 * page 2, line 5 - page 2, line 9 * * page 4, line 35 - page 16, line 12 *	1,2,5,6	TECHNICAL FIELDS SEARCHED (Int.Cl.6) G07F H04L G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	29 June 1999	Aupiais, B

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 95 10 6967

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

29-06-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9309620 A	13-05-1993	AU 2883692 A	07-06-1993
		DE 69218961 D	15-05-1997
		DE 69218961 T	24-07-1997
		EP 0611506 A	24-08-1994
		ES 2101124 T	01-07-1997
		HK 1000987 A	15-05-1998
		SG 44714 A	19-12-1997
		US 5479511 A	26-12-1995
US 4748668 A	31-05-1988	AT 81414 T	15-10-1992
		AU 592207 B	04-01-1990
		AU 7526687 A	14-01-1988
		DE 3782099 A	12-11-1992
		EP 0252499 A	13-01-1988
		JP 2511464 B	26-06-1996
		JP 63101987 A	06-05-1988
US 5121431 A	09-06-1992	CA 2045385 A,C	03-01-1992
US 5299263 A	29-03-1994	CA 2157011 A	15-09-1994
		EP 0691055 A	10-01-1996
		JP 8507619 T	13-08-1996
		WO 9421067 A	15-09-1994
		US 5406628 A	11-04-1995
EP 0502712 A	09-09-1992	JP 4276787 A	01-10-1992
		JP 4277789 A	02-10-1992
		JP 5068032 A	19-03-1993
		US 5313530 A	17-05-1994
		EP 0531158 A	10-03-1993
		US 5321752 A	14-06-1994

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82